Validation, Verification and Evaluation of Visualization Software: Position Statement

Al Globus, CSC at NASA Ames

I want to make two points: 1. visualization software needs rigorous verification in the form of much better testing and, 2. experiments with human subjects are essential to scientifically validate and evaluate visualization techniques.

Verification - does the software do what the developer thinks it does?

I've found much visualization software to be pretty buggy. Crashes and mysterious behavior are common. Only when I've used a package for some time and know it quite well can I get reliable results. What's a developer to do?

I've tried a few things not specific to visualization, adding a 'static test()' to each C++ class, 'hand simulating' by single stepping through the code using a visual debugger watching all local variables and object members update, and building a random widget tweaker. I've also tried developing a test set generator for unsteady flows. The first two are pretty obvious and fairly standard, although (apparently) rarely done. The last two merit some discussion.

A random widget tweaker keeps a list of all of the user input widgets. In overnight runs, the tweaker repeatedly choose a widget at random and send it the tweak message. The tweak message changes the value of a widget (e.g., slider position) at random. This technique simulates a monkey at the keyboard and mouse. It effectively finds crash and burn bugs.

Building a good test set generator is an interesting problem. First of all, unsteady data sets can be very large, so distributing the source to a test set generator saves a lot of network bandwidth. Output size can be a parameter to the code so that small data sets can be generated for debugging and larger sets (that just fit currently available disk space) generated to investigate program performance. Test set should reveal common and subtle bugs and deficiencies that generate incorrect pictures. For example, circular streamlines will stress some particle tracing codes.

Validation - does the visualization accurately, and effectively, represent the data?

Many visualization programmers come from the computer graphics community, as I do. This community values pretty pictures; which are not necessarily correct or informative. In many cases, visualizations are accepted if they look 'more or less right'. Sometimes a user is called in to glance at the visualization and make a few comments. This is mediocre science, at best.

We claim that visualization increases human understanding. This can only be proven by experiments with human subjects. As far as I know, no such experiments have ever been conducted. Such experiments are difficult to design and so require collaboration with psychologists and/or human factors experts.

Evaluation - is thing A better than thing B?

When is one visualization techniques better than another? We can flame or run experiments. For example, two groups of subjects are given a data set and asked to find important features. Each group is given a different visualization tool (e.g., isosurfaces *vs.* scalar mapped cut planes). Time to completion and correct results are measured.

Biography

Al Globus is a senior computer scientist with Computer Sciences Corporation at NASA Ames Research Center. His research interests include scientific visualization, space colonization, and computer network enhanced education.

Globus received a B.A. degree in information science from the University of California at Santa Cruz in 1979 following a previous life as a musician. He is a member of the IEEE Computer Society and the American Institute of Aeronautics and Astronautics.